SDC Localization Competition

Introduction

- The goal of this competition is to develop a localization module for estimating the poses of a self-driving car given a map.

- Datasets:
  1. NuScenes dataset
  2. ITRI dataset

Challenges

- There will be 2 different level scenes(easy and medium) for the localization.

- We offer a basic localization approach with ICP scan matching as the baseline. You are free to use any approaches the do the localization.

  If you don’t have any idea about the localization, have a look at the document.

- Your algorithm(s) need to have a better performance than our baseline or you won’t get any credit.

- You also need to compare your performance with other teams to get more credit.

Challenges

- **Test**

  NCTU campus (Just for practice)

- **Easy**

  ITRI campus (Available in 4/22)

- **Medium**

  Urban from Nuscenes (Available in 4/29)

Dataset detail

- ITRI campus

LiDAR:	/lidar_points (sensor_msgs/PointCloud2)
IMU:	/imu/data (sensor_msgs/Imu)
GPS:	/fix (geometry_msgs/PointStamped)
Transformation of sensors:	/tf (tf2_msgs/TFMessage)

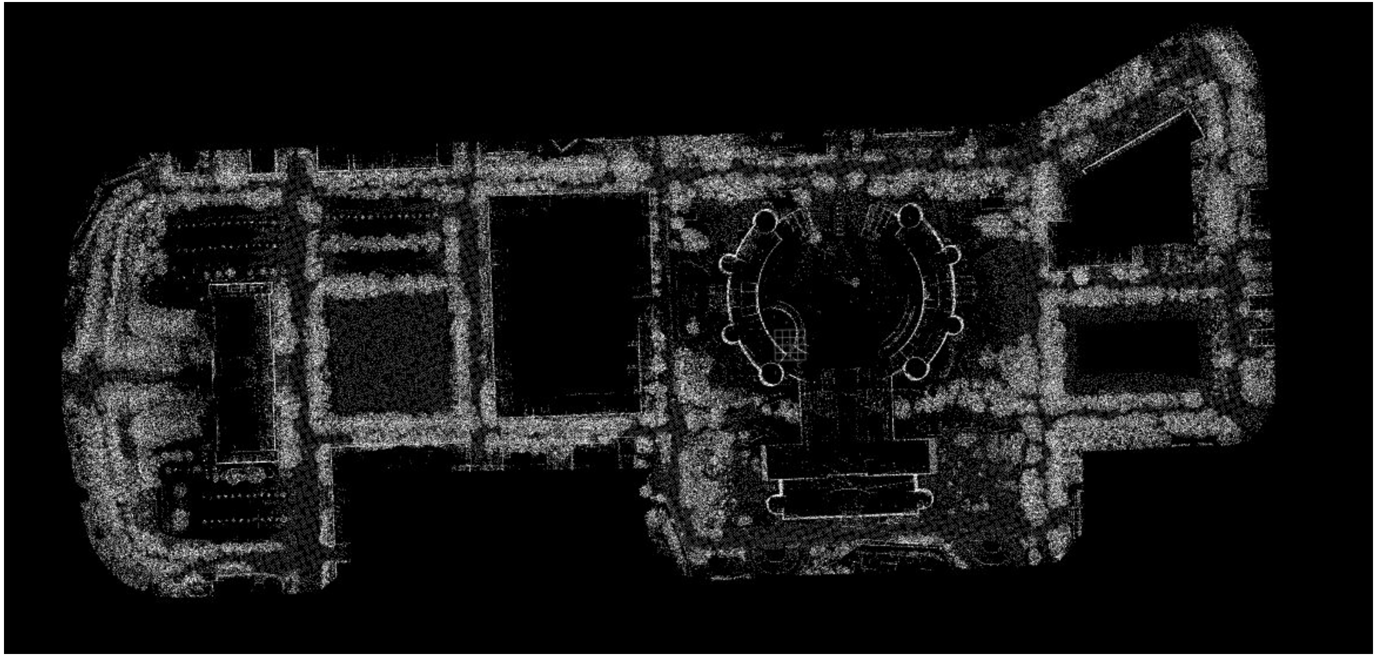
Dataset detail

- ITRI campus

LiDAR:	velodyne VLP-32C
IMU:	xsens MTi-G-710
GPS:	simulated from vehicle pose (standard deviation of noise: 1 meter)

Dataset detail

- ITRI campus



Submission rules

- In easy and medium level, we provide two types data for you. The public data just for you to try and the private data is for evaluation.
- Public data:

  The data **with** ground truth for a few seconds. You can test your algorithm(s) with these data via the evaluated program by yourself.

- Private data:

  The data **without** ground truth for a few seconds. We will evaluate the final ranking with the private data. You can keep uploading your result from private data once a day until the deadline, We will evaluate and rank all the team.

Results format

- The file of your result need to be csv file which named “{teamnumber_case_bagnumber}.csv”. For example: 2_easy_2.csv

- In the end of this competition, all teams need to upload the codes. We will check if the code can be compiled and executed.

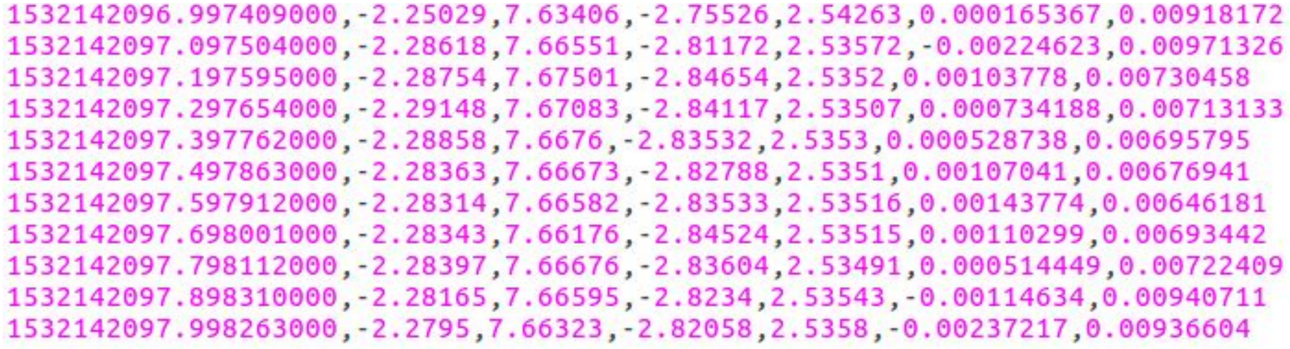
- The result should contains coordinate data for every LiDAR timestamp. If the result lost any timestamp, your would get 0 score for that timestamp.

Results format

- The result contains the pose data of each timestamp like:

  (timestamp, x , y, z, yaw, pitch, roll).

- Your result should look like this:



```
1532142096.997409000,-2.25029,7.63406,-2.75526,2.54263,0.000165367,0.00918172
1532142097.097504000,-2.28618,7.66551,-2.81172,2.53572,-0.00224623,0.00971326
1532142097.197595000,-2.28754,7.67501,-2.84654,2.5352,0.00103778,0.00730458
1532142097.297654000,-2.29148,7.67083,-2.84117,2.53507,0.000734188,0.00713133
1532142097.397762000,-2.28858,7.6676,-2.83532,2.5353,0.000528738,0.00695795
1532142097.497863000,-2.28363,7.66673,-2.82788,2.5351,0.00107041,0.00676941
1532142097.597912000,-2.28314,7.66582,-2.83533,2.53516,0.00143774,0.00646181
1532142097.698001000,-2.28343,7.66176,-2.84524,2.53515,0.00110299,0.00693442
1532142097.798112000,-2.28397,7.66676,-2.83604,2.53491,0.000514449,0.00722409
1532142097.898310000,-2.28165,7.66595,-2.8234,2.53543,-0.00114634,0.00940711
1532142097.998263000,-2.2795,7.66323,-2.82058,2.5358,-0.00237217,0.00936604
```

Evaluation metrics

Below we define the metrics for the our localization task. Our final score is a weighted sum of root-mean-square (RMS) of translation score and rotation score.

$$\begin{aligned} FinalScore = 0.4 \times TranslationScore_{RMS} + 0.4 \times YawScore_{RMS} + \\ 0.1 \times PitchScore_{RMS} + 0.1 \times RawScore_{RMS} \end{aligned}$$

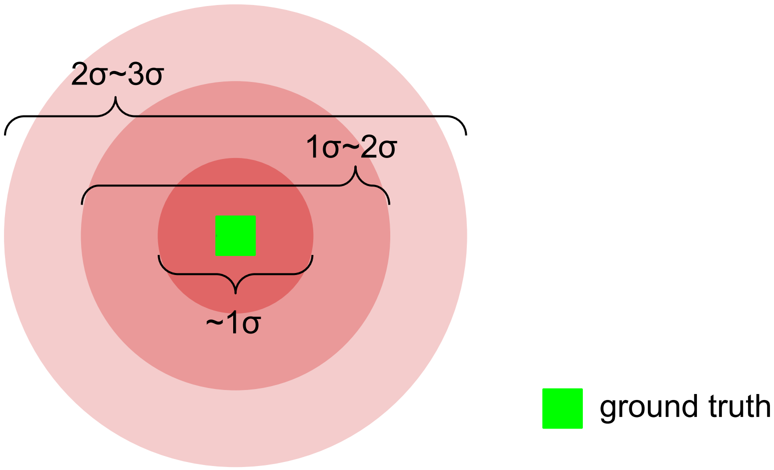
Translation score RMS

We define the translation score RMS which is calculated by the L2 distance between your result and ground truth in three dimensional Euclidean space. The σ is the covariance of the ground truth.

Score for each scan:

- ~1σ: 1
- 1σ~2σ: 0.7
- 2σ~3σ: 0.3
- 3σ~: 0

$$TranslationScore_{RMS} = \sqrt{\frac{\sum_{i=0}^{n} score_i^2}{n}}$$



Rotation score RMS

- There are many ways of expression about the rotation like Quaternion, Matrix,....
- Here we define the score with yaw, pitch, roll form(radians) and compute them **independently**.

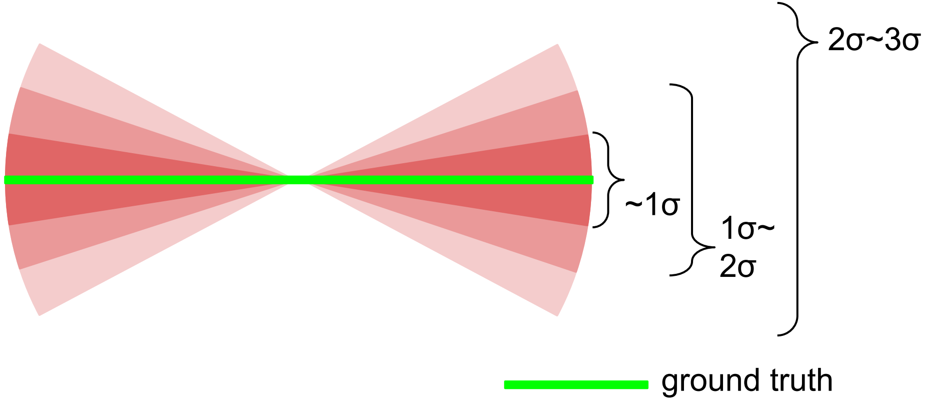
Rotation score RMS

- We define the rotation score RMS which is calculated by the error between your result and ground truth in three dimensional Euclidean space. The σ is the covariance of the ground truth.

- Score for each scan:

- ~1σ: 1
- 1σ~2σ: 0.7
- 2σ~3σ: 0.3
- 3σ~: 0

$$RotationScore_{RMS} = \sqrt{\frac{\sum_{i=0}^{n} score_i^2}{n}}$$



Ranking and Grading

- Competition Ranking: 60%
- Presentation: 40%

Competition Ranking

- Easy
  - Top 10%: 50
  - Top 25%: 45
  - Top 50%: 40
  - Others: 35
  - Below baseline: 0
- Medium
  - Top 10%: 50
  - Top 25%: 45
  - Top 50%: 40
  - Others: 35
  - Below baseline: 0

Presentation

- Proposal: 10
- Report: 20
- Contribution: 30
- Final Presentation: 40

The contribution part is depends on the idea you implement or how you solve the issue you faced.

If you use whole open source project without your idea, you will get zero in the contribution part.

Download

- Dataset and evaluated program